



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/756,894	01/13/2004	John Randall West	07464-0002	3627
25213	7590	10/06/2006	EXAMINER PHAM, THAI V	
HELLER EHRMAN LLP 275 MIDDLEFIELD ROAD MENLO PARK, CA 94025-3506			ART UNIT 2191	PAPER NUMBER

DATE MAILED: 10/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/756,894	WEST ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Thai Van Pham	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 13 January 2004.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1 - 49 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1 - 49 is/are rejected.
- 7) ☒ Claim(s) 2 and 49 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 January 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>05/10/2006</u> . | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

This is the initial office action based on the application filed on September 29, 2006.

Claims 1 – 49 are currently pending and have been considered below.

#### **Specification**

1. The disclosure is objected to because of the following informalities:

-- Paragraph [0036] in the specification refers to "*flow control structure 32*" which is not identified anywhere in the drawings.

-- Paragraph [0043] in the specification refers to "*an identification field 56*" which is not identified anywhere in the drawings.

-- The last sentence of paragraph [0043] in the specification refers to "*the same application object 16*" which should correctly be identified as "... 24".

-- Paragraph [0045] in the specification use the same number "122" to identify both "*logic 122*" and "*an abstract test case representation 122*".

-- Paragraph [0045] in the specification use the same number "126" to identify both "*logic 126*" and "*an application metadata repository 126*".

Appropriate corrections are required.

#### **Drawing**

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: "32" and "56" (See objections to Specification above.) Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to

avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character:

-- "54" has been used to identify both "*Application Meta Data Repository*" in Figure 4 and "*Application Event Effect*" in Figure 6.

-- "381" in Figure 11 is incorrectly labeled – it should be "318" as referred to in paragraph [0054] in the specification.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Appropriate corrections are required.

***Claim Objections***

4. Claims 2 and 49 are objected to because of the following informalities:

-- Claim 2 refers to itself as the parent claim – “*method of claim 2*”.

-- Claim 49 does not contain a claim-ending period.

Appropriate corrections are required.

***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1 – 49 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

-- Claims 1 – 49: cite simply a method and a system for transforming test cases into a higher abstract. The claimed invention is not practical due to its lack of a useful and tangible result. Thus, the claimed method and system are non-statutory according to 35 U.S.C 101.

-- Claim 49: cites a carrier medium as the claimed subject matter. Since the disclosure provides no explicit definition for the term carrier medium, a carrier medium is given its broadest reasonable interpretation. A carrier medium may be in form of any physical computer-readable storage devices or electromagnetic field. While a computer-readable storage device is directed to a statutory subject matter, electromagnetic is not

Art Unit: 2191

because it is a natural phenomenon. As a result, the subject matter directed to in the claim is non-statutory according to 35 U.S.C 101.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 14 and 15 are rejected due to lacking sufficient antecedent bases for the limitations in the claims:

-- Claim 14 recites the limitation "*the syntax analysis*" which is not mentioned in its parent claim(s).

-- Claim 15 recites the limitation "*the abstract syntax tree*" which is not mentioned in its parent claim(s).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1 – 10, 30 – 38, 42, 43, and 47 – 49 are rejected under 35 U.S.C. 103(a) as being obvious over **Rosaria** (6,976,246).

-- Claims 1 and 49:

**Rosaria** discloses *a method and a computer readable medium tangibly containing a program for executing a method, the method comprising:*

- *importing test cases written in one or more scripting languages* (Fig. 3, page 6: lines 38 – 65; Fig. 6, page 9: line 16 – page 10: line 56; “model editor” and “rules editor” receiving user-input data);
- *converting test cases to an abstract representation that includes application state, external interaction sequences and input data* (Fig. 3, page 6: lines 18 – 27; finite state model and associated attributes and transitional conditions).

**Rosaria** does not explicitly disclose that the method further comprising

- *storing abstract representation of test cases into a database system.*

He, however, specifically discloses storing data of test case models as well as intermediate and executable files of the test case generator program in volatile and nonvolatile memories of the system (Fig. 4, page 7: line 32 – page 8: line 46). Official notice is taken that database systems had been widely known and used to logically

organize related data to ensure its integrity and quality at the time of the invention was made. Thus, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a database system for storing the data of test case models in order to provide better data organization and manageability for the program application.

-- Claims 38 and 43:

**Rosaria** discloses *a system for transforming test cases, comprising:*

- *a processor for importing test cases written in one or more scripting languages (Fig. 3, page 6: lines 38 – 65; Fig. 6, page 9: line 16 – page 10: line 56; “model editor” and “rules editor” receiving user-input data);*
- *logic for converting test cases to an abstract representation that includes application state, external interaction sequences and input data (Fig. 3, page 6: lines 18 – 27; finite state model and associated attributes and transitional conditions).*

**Rosaria** does not explicitly disclose that the system further comprising

- *a database for storing abstract representation of test cases.*

He, however, specifically discloses storing data of test case models as well as intermediate and executable files of the test case generator program in volatile and nonvolatile memories of the system (Fig. 4, page 7: line 32 – page 8: line 46). Official notice is taken that database systems have been widely known and used to logically organize related data to ensure its integrity and quality at the time of the invention was made. Thus, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a database system for storing the data of test case



models in order to provide better data organization and manageability for the program application.

-- Claim 2:

**Rosaria** discloses *the method of claim 1*,

- *wherein an application state represents a runtime snapshot of application under test which defines the context of external interaction (Fig. 3, page 6: line 66 – page 7: line 11; a state in the finite state model).*

-- Claim 3:

**Rosaria** discloses *the method of claim 2*,

- *wherein the application state includes a set of application objects, its attributes and attribute values (Fig. 3, page 6: lines 38 – 65; Fig. 7, page 10: line 57 – page 11: line 6; “a state” contains objects of the state model and their attributes as specified in “rules” setting).*

-- Claim 4:

**Rosaria** discloses *the method of claim 2*,

- *wherein the applications states corresponding to a test case are arranged in a hierarchical manner (Figs. 2 and 9, page 13: line 31 – page 14: line 55; states and their dependencies).*

-- Claims 5 and 6:

**Rosaria** discloses *the method of claim 2* but does not explicitly disclose that

- *the database system is a relational database management system or an XML database management system.*

Official Notice is taken that a relational database management system or XML database system had been among the well-known conventional models of database system in the art at the time the invention was made. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to implement the database system using a relational database management system or an XML database management system to provide the method with a standardized interface and protocol for storing data.

-- Claim 7:

**Rosaria** discloses *the method of claim 2* but does not explicitly disclose that

- *the scripting languages can be typed or un-typed programming languages used for recording or authoring test cases.*

Official Notice is taken that typed programming languages such as Pearl and Tcl as well as un-typed programming languages such as DOS and UNIX Shells commands had been well known and used in scripting at the time the invention was made. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made that to use one or a mixture of typed and un-typed languages for the scripting language to provide the user with ease and comfort in understanding and writing test scripts.

-- Claims 8 and 9:

**Rosaria** discloses *the method of claim 2,*

- *wherein the external interaction sequences represent events invoked by external agents on the application objects and the external agents can be either human agents or other software agents (Fig. 3, page 6: lines 18 – 27; state transitional conditions implies that events that are defined using “model editor” and “rules editor” to cause a state transition are, in fact, interaction sequences that can be external to an application object, which are events triggered by external agents associated with the application itself).*

-- Claim 10:

**Rosaria** discloses *the method of claim 8,*

- *wherein the interaction sequencing includes flow control structures for capturing sequential, concurrent, looping and conditional interactions (Fig. 8, page 11: line 7 – page 13: line 30; transitional operators and state flow control in rules editor).*

-- Claims 30 – 33, and 42/47:

**Rosaria** discloses *the method of claim 2 (and the system of claim 38/43) further comprising (and the system further comprising logic for):*

- *enriching the abstract representation of test cases with information from an application metadata repository (Figs. 3, 9, and 10; “data structure” and “operation modes” of states, “graph traversal program”);*

- *wherein the enrichment of abstraction representation of test cases involves extracting values for those attributes of application objects associated with the test cases that are missing in the incoming test scripts (Fig. 10; page 14: line 62 – page 15: line 8; “graph traversal program” for designating test algorithm used in generating test sequence);*
- *wherein enriching the abstraction representation of test cases includes decoupling of test cases from their recording or authoring environments (Fig. 10; page 14: line 62 – page 15: line 8; “graph traversal program” are separate from “model editor” and “rules editor”); and*
- *wherein enriching the abstraction representation of test cases allows usage of attributes that are stable within an application metadata representation (Fig. 10; page 14: line 62 – page 15: line 8; applying “graph traversal program” to state model).*

-- Claims 34 – 36:

**Rosaria** discloses *the method of claim 33,*

- *wherein using an identification field for a given object within the application metadata repository improves the reusability of a test case instead of a label used to represent the same object within a user interface which can change based on the locale of the application (Fig. 6; “operation modes”);*
- *wherein using an identification field allows to overcome the problem of different test execution environments using different attributes to identify the same application object (Figs. 6 and 8; “operation modes” in “model editor” and “rules editor”); and*

- *wherein enriching the abstraction representation of test cases enables representation of test cases that are test execution environment independent (page 4: line 66 – page 5: line 23).*

-- Claims 37 and 48:

**Rosaria** discloses *the method of claim 2 (and the system of claim 43) further comprising (and the system further comprising logic for):*

- *separating application object attributes and input data from external interaction sequencing provides automatic parameterization (page 6: line 38 – 65; state and operation mode parameterization).*

8. Claims 11 – 29, 39 – 41, and 44 – 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Rosaria** (6,976,246) in view of **Venter** (2004/0194072).

-- Claims 11 – 14, and 39/44, 45:

**Rosaria** discloses *the method of claim 2 (and the system of claim 38/43) but does not explicitly disclose that the method further comprising (and the system further comprising logic for):*

- *implementing a syntax analyzer for incoming scripts;*
- *wherein the syntax analyzer is implemented one for each scripting language;*
- *wherein the syntax analyzer utilizes rules of syntax analysis that are specified in Extended Backus-Naur Form (EBNF), and*

- *wherein the syntax analysis generates a parse tree in the form of an Abstract Syntax Tree (AST).*

**Venter** discloses a method for

- *implementing a syntax analyzer for incoming scripts* (Fig. 2, [0031] – [0032]; “primary syntax analyzer” and “several secondary syntax analyzers”);
- *wherein the syntax analyzer is implemented one for each scripting language* (Fig. 2, [0031] – [0032]; “primary syntax analyzer” and “several secondary syntax analyzers” corresponding to the languages being supported)
- *wherein the syntax analysis generates a parse tree in the form of an Abstract Syntax Tree (AST)* (Figs. 3 – 4; [0035] – [0038]; “primary syntax analyzer” creates an “Abstract Syntax Tree (AST)” and “the secondary syntax analyzers” subsequently modify the AST in accordance to the program structure).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to realize that the model-based testing method of **Rosaria** implements a syntax analyzer for parsing the test script where the syntax analyzer is specifically compatible with the scripting language, and the syntax analyzer generates an AST. A syntax analyzer is necessary in translating any kind of scripting or graphically captured language into a lower-level intermediate or computer executable language, while an AST is commonly a result generated by a syntax analyzer.

- Official Notice is taken that EBNF had been well known as a standardized set of syntax analysis rules at the time the invention was made. Thus, it would also have

been obvious to recognize that EBNF can be used for syntax analysis by the syntax analyzer in the invention of **Rosaria** in view of **Venter**.

-- Claims 15, 16, and 40/46:

**Rosaria** discloses *the method of claim 2 (and the system of claim 38/43)* but does not explicitly disclose that the method *further comprising (and the system further comprising logic for):*

- *implementing a semantic analysis that converts the abstract syntax tree to an abstract test case representation based on an Application Object Model (AOM);*
- *wherein the semantic analysis decomposes the test cases represented as an Abstract Syntax Tree into application state, external interaction sequences and input data.*

**Venter** discloses a method for

- *implementing a semantic analysis that converts the abstract syntax tree to an abstract test case representation based on an Application Object Model (AOM) ([0040]; the “primary compiler” includes “a semantic analyzer” that checks the source code and AST for semantic errors);*
- *wherein the semantic analysis decomposes the test cases represented as an Abstract Syntax Tree into application state, external interaction sequences and input data ([0040]; “the semantic analyzer” checks the source code and AST for semantic errors and subsequently converts the AST to “visitor” classes which contain the corresponding attributes of the state models defined in the test case models)).*

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to realize that the method of **Rosaria** further implements a

semantic analyzer for semantically analyzing and converting the AST generated by the syntax analyzer into an intermediate representation with the associated state model user-defined attributes.

-- Claims 17, 18, and 22:

**Rosaria** and **Venter** disclose *the method of claim 15*, and

**Rosaria** further discloses that

- *an application object model is a metadata representation for modeling application under test; wherein the metadata representation includes object type definitions for application objects; and wherein application object type definitions include additional categorization of each application object types into hierarchical, container and simple types (Fig. 9, page 13: line 31 – page 14: line 55; data structure of states and their attributes, and dependencies between states).*

-- Claims 23 – 25:

**Rosaria** and **Venter** disclose *the method of claim 22*, and

**Rosaria** further discloses that

- *the hierarchical object types are associated with an application state of its own (Figs. 2 and 9; independent states within a state model);*
- *wherein application object types that can contain instances of other objects are termed as container types (Figs. 2 and 9; states that are dependent upon each other).*
- *the state associated with a hierarchical application object type is a modal application state or a non-modal application state; wherein a modal application state restricts*



*possible interactions to application object instances available within the current application state* (Fig. 2 and 9; types of “states” in a FSM, such as Mealy and Moore types.)

-- Claims 26 – 29, 41:

**Rosaria** and **Venter** disclose *the method of claim 22* (and *the system of claim 38*), and **Rosaria** further discloses that

- *the effects of events on an application state capture one or more consequences of the event to the application state* (Figs. 2 and 8, page 11: line 7 – page 13: line 30; “state transitions” caused by external/internal events);
- *wherein a consequence of an event is selected from, creation of a new object instance of a given type, deletion of an object instance of a given type, modification of attributes of an existing object instance and selection of an instance of an object type* (Fig. 8, page 11: line 7 – page 13: line 30, “rules” defining program flow of “state model”);
- *wherein creation of a new instance of an object of type that is hierarchical results in creation of a new application state* (Fig. 8, page 11: line 7 – page 13: line 30, dependent states); and
- *wherein selection of an object instance of type that is hierarchical results in selection of the application state associated with that object instance* (Fig. 8, page 11: line 7 – page 13: line 30; states and their dependencies as characterized according to the application structure and properties).

-- Claims 19 – 21:

**Rosaria** and **Venter** disclose *the method of claim 17*, and

**Rosaria** further discloses that

- *the metadata representation includes attribute definitions for each application object type* (Fig. 7, page 10: line 57 – page 11: line 6; Fig. 9, page 13: line 31 – page 14: line 55; state attributes in data structures);
- *wherein the metadata representation includes definition of methods and events that are supported by each application object type* (Fig. 7, page 10: line 57 – page 11: line 6; Fig. 9, page 13: line 31 – page 14: line 55; “operation modes and state attributes in data structures); and
- *wherein the metadata representation includes definition of effects of events on an application state* (Fig. 7, page 10: line 57 – page 11: line 6; Fig. 9, page 13: line 31 – page 14: line 55; data structures of states and operation modes).

***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

-- **Kossatchev et al.** (6,698,012), Method and System for Testing Behavior of Procedures. Nortel Networks Limited.

-- **Srivastava et al.** (6,609,248), Cross Module Representation of Heterogeneous Programs, Microsoft Corporation.

-- **Jean S. Hartman et al.** (6,505,342), System and Method for Functional Testing of Distributed Component-Based Software. Siemens Corporate Research Inc..

-- **Parker et al.** (5,600,789), Automated GUI Interface Testing. Segue Software, Inc..

Art Unit: 2191

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thai Van Pham whose telephone number is (571) 270-1064. The examiner can normally be reached on Monday - Thursday, 9am - 5pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TVP

TVP

WEI ZHEN  
SUPERVISORY PATENT EXAMINER



Wei Y. Zhen  
Supervisory Patent Examiner